Benchmark Environments for Multitask Learning in Continuous Domains

{ Peter.Henderson, Wei-Di.Chang, Florian.Shkurti, Johanna.Hansen, David.Meger, Gregory.Dudek } @mail.mcgill.ca

Introduction

As demand drives systems to generalize to various domains and problems, the study of multitask, transfer and lifelong learning has become an increasingly important pursuit. In discrete domains, performance on the Atari game suite has emerged as the *de facto* benchmark for assessing multitask learning. However, in continuous domains there is a lack of agreement on standard multitask evaluation environments which makes it difficult to compare different approaches fairly.

In this work, we describe a benchmark set of tasks that we have developed in an extendable framework based on OpenAI Gym. We run a simple baseline using Trust Region Policy Optimization and release the framework publicly to be expanded and used for the systematic comparison of multitask, transfer, and lifelong learning in continuous domains.

Description of Environments

We provide a number of environments including those which are generalized modifications of standard gym environments as well as novel continuous domains. Our initial release investigates adding flexibility to standard OpenAI gym MuJoCo environments: modifying gravity, adding sensor readouts and a random wall obstacle, perturbing body-part sizes, and adding random goal/start state positions for arm environments. We also add an original set of environments for learning policies in continuous navigation tasks.

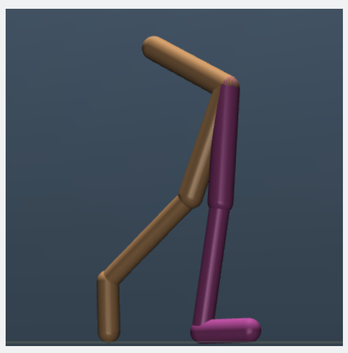

Figure 1: Example Mujoco multitask environments: Walker2d with small feet

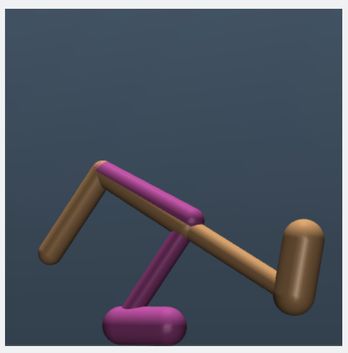

Figure 2: Example Mujoco multitask environments: Walker2d with large feet

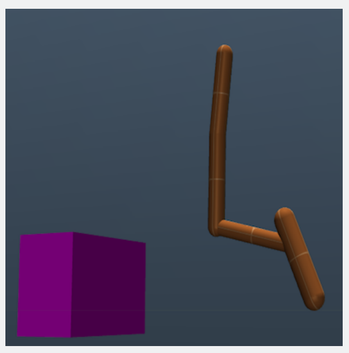

Figure 3: Example Mujoco multitask environments: Hopper with wall obstacle

Mujoco modified environments -
Based on running and arm-based OpenAi Gym environments

- Varying gravity for running agents
- Varying scales of mass/width for body parts (thigh, feet, etc)
- Randomly move start/goal positions for robot arm tasks
- Combine multiple tasks (rewards for standing up and running) in humanoid environments

2D Navigation environments -
Novel environments which require agent to travel to a goal.

- Image-based environment - agent has access to map, position, goal
- State-based environment - agent has access to position, and range-and-bearing to nearest obstacle
- Navigation based environment - agent only has access to range-and-bearing data using ray-tracing (needs SLAM)

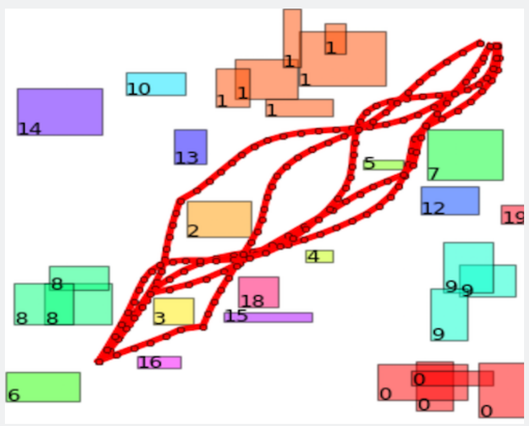

Figure 4: Example state-based navigation environment

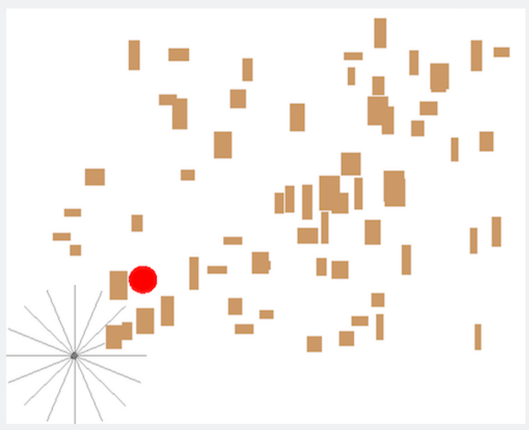

Figure 5: example range-and-bearing navigation environment

Pull Requests Welcome

We actively maintain a Github repository and website detailing experiments. We provide a location for new environments and benchmark experiment results to be provided to the community on top of our existing infrastructure.

http://github.com/Breakend/gym-extensions

References

[1] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[2] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

[3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1889–1897, 2015.

Multitask Environments

We develop several sets of intuitive task groups which can serve as simple benchmarks which increase in complexity both within the group and in our listing order. We introduce the following environment groups:

- Modified environments with different gravity parameters[a]
- Modified environments with sensor readouts (simply reading zero if no wall) and permuted with a random wall in the runner path
- The OpenAI Gym Striker environment with both random start position of the object as well as random goal state
- The OpenAI Gym Pusher environment with both random start position of the object as well as random goal state
- Humanoid model: Learning to standup and run; learning to standup, run, and jump over walls
- Learning to run with different sized limbs with the base set of limbs encompassing {Torso, Leg, Thigh, Foot} and specific extra limbs listed below (i.e. example combinations look like: HumanoidBigArm-v0, HopperSmallFoot-v0).
- Learning to navigate and search in 2D environments using only current position and distance to closest obstacles (State-Based-Navigation-2d-Map{0-9}-Goal{0-2}-v0)
- Learning to navigate and search in 2D environments observing current position, distance to closest obstacles, and known goal position (State-Based-Navigation-2d-Map{0-9}-Goal{0-2}-KnownGoalPosition-v0)
- Learning to navigate and search in 2D environments observing only raytracing distance readouts (Limited-Range-Based-Navigation-2d-Map{0-9}-Goal{0-2}-v0)
- Learning to navigate and search in 2D environments observing current position, raytracing distance readouts, and known goal position (Limited-Range-Based-Navigation-2d-Map{0-9}-Goal{0-2}-KnownPositions-v0)
- Learning to navigate and search in 2D environments observing only the 2D map image with goal location and current position highlighted in different colors (Image-Based-Navigation-2d-Map{0-9}-Goal{0-2}-v0)

[a] {BaseRunningEnv} denotes one of the OpenAI Gym environments from: Humanoid, Hopper, Walker2d, HalfCheetah with {GravityVariation} from {Half, ThreeQuarters, OneAndQuarter, OneAndHalf}.

Example Benchmark using TRPO

The table below illustrates two of our proposed multitask benchmarks groups, Modified Gravity Hopper (top) and Modified Gravity Walker2d (bottom). Each of our groupings presents five slightly modified environments, listed by increasing complexity, from which we can understand a policy's generalized performance. Results are listed in terms of average and standard deviation ($\mu \pm \sigma$) of reward over 20 sample rollouts obtained by running the RLLab [1] implementation of Trust Region Policy Optimization [3] (TRPO) using an identical policy network to [2].

Hopper				
Gravity	**Fully Trained**	**After Env Training**	**First Step**	**Single Env**
0.5	1495.93 ± 823.51	2352.19 ± 580.53	13.48 ± 8.71	1843.89 ± 485.25
0.75	413.77 ± 252.67	2245.13 ± 872.16	697.96 ± 210.79	2328.09 ± 834.35
1.0	668.52 ± 159.90	2622.31 ± 1032.45	781.88 ± 262.35	3232.87 ± 582.55
1.25	922.76 ± 128.71	3006.17 ± 847.30	818.08 ± 255.85	3028.04 ± 875.81
1.50	2690.57 ± 1110.39	2792.72 ± 1075.30	658.15 ± 117.14	2169.07 ± 825.75
Total Grouping	990.95 ± 1022.32	2603.704 ± 881.54	593.91 ± 184.43	2520.39 ± 720.74

Walker2d				
Gravity	**Fully Trained**	**After Env Training**	**First Step**	**Single Env**
0.5	1366.07 ± 1126.59	3485.19 ± 1054.06	5.35 ± 10.30	2231.86 ± 902.31
0.75	3686.37 ± 287.96	3962.69 ± 1061.71	1071.95 ± 267.35	2431.87 ± 935.14
1.0	4030.00 ± 85.76	3732.04 ± 1314.89	930.92 ± 264.88	2570.15 ± 915.58
1.25	4115.23 ± 90.33	4090.30 ± 1058.62	926.06 ± 303.76	3505.52 ± 1626.58
1.50	4201.08 ± 684.37	3988.62 ± 971.43	925.93 ± 290.33	2435.21 ± 1391.00
Total for Grouping	3479.76 ± 1230.72	3851.768 ± 1092.1	772.04 ± 227.32	2634.92 ± 1154.12

Results in the **Fully Trained** column were obtained on the row's environment after a policy had been fully trained on all environments in its group. The **After Env Training** columns demonstrates performance after training on the specific row's environment after training only on previous environments within its group, with training performed in ascending order within the group. The **First Step** column indicates the reward at the very first iteration of training on the row's environment after having trained on previous environments in the group. **Single Env** results were found by training solely on the row's environment, with no exposure to other environments in the group. Hyperparameters were held constant throughout each of the tasks within the group. While this isn't an explicit multitask learning approach, this provides basic insights (using a well-known reinforcement learning algorithm) into forward transfer and generalization of a policy on these task groupings. The two group results in the table are representative of experiments performed in the paper, and allow us to demonstrate the following:

- Environments with more unstable dynamics (such as Hopper) tend to demonstrate catastrophic forgetting in sequential learning over different environments (perform better in After Env Training than Fully Trained).
- More stable environments (such as Walker2d) tend to demonstrate improved learning over varying environment when compared to learning on a single environment (perform better in Fully Trained than Single Env).

Future Work

In future releases we also plan to add standard environments for adding motor noise, arm environments where the end-goal position has a velocity (such that the arm must track the target), and making the sensor-based environments more realistic (and thus more transferable to real-world systems).